

Intro: Welcome to the CS61BL Midterm!

Your name: Jiayi Xu

Login: su22-s

Your SID: 3037100228

[In Person] Login of person to your Left: \_\_\_\_\_ Right: \_\_\_\_\_

Lab TA: Ryan Miqui

Many of the problems on this exam consist of multiple choice or fill in the blank questions. Make sure to read the directions carefully!

There are also a few code writing questions. You may not need to use all lines provided. If you are given skeleton code to fill in, we will not grade your answer if you alter the skeleton code. If you are given a line limit, we will not grade your answer if you go over the line limit or if you don't properly format your code in the attempt of using fewer lines. For instance, the properly formatted code below takes 9 lines, counting the function signature and closing brace. **You may not use ternary operators.**

```
1 public static boolean function() {  
2     for (int i = 0; i < 10; i += 2) {  
3         if (i == 2) {  
4             i -= 1;  
5         } else {  
6             i += 1;  
7         }  
8     }  
9 }
```

However, the code below is **not**, as there are a number of mistakes!

```
1 public static boolean function() {  
2     for (int i = 0; i < 10; i += 2) {  
3         if (i == 2) { i -= 1; } //We will count this conditional as 3 lines!!  
4         else { i += 1; } }  
5 }
```

Additionally, we will not grade your question if you fail to follow any restrictions given in the problem statement. Your code won't be checked by a compiler, but we will take off points for errors that are more than a typo. Please pay attention to detail when answering coding questions!

## Reference Sheet

Note that not everything on this reference sheet may be needed.

### System.arraycopy

```
// Copies length elements from src starting at srcPos to dest starting at destPos
// int[] can be replaced by an array holding any type
System.arraycopy(int[] src, int srcPos, int[] dest, int destPos, int length)
```

### JUnit Methods

```
assertEquals(Object expected, Object actual)
assertEquals(int expected, int actual)
assertEquals(double expected, double actual)
assertTrue(boolean actual)
assertFalse(boolean actual)
assertNotNull(Object actual)
assertArrayEquals(Object[] expected, Object[] actual)
assertArrayEquals(int[] expected, int[] actual)
assertArrayEquals(double[] expected, double[] actual)
```

### SLList

```
public class SLList {
    public class IntNode {
        public int item;
        public IntNode next;
        public IntNode(int i, IntNode n) {
            item = i;
            next = n;
        }
    }
    private IntNode first;
    public SLList(int x) {
        first = new IntNode(x, null);
    }
    public void addFirst(int x) {
        first = new IntNode(x, first);
    }
}
```



## Iterator, Iterable, Comparator, and Comparable

```
public interface Iterator<T> {  
    boolean hasNext();  
    T next();  
}
```

```
public interface Iterable<T> {  
    Iterator<T> iterator();  
}
```

```
public interface Comparator<T> {  
    int compare(T o1, T o2);  
}
```

```
public interface Comparable<T> {  
    int compareTo(T obj);  
}
```

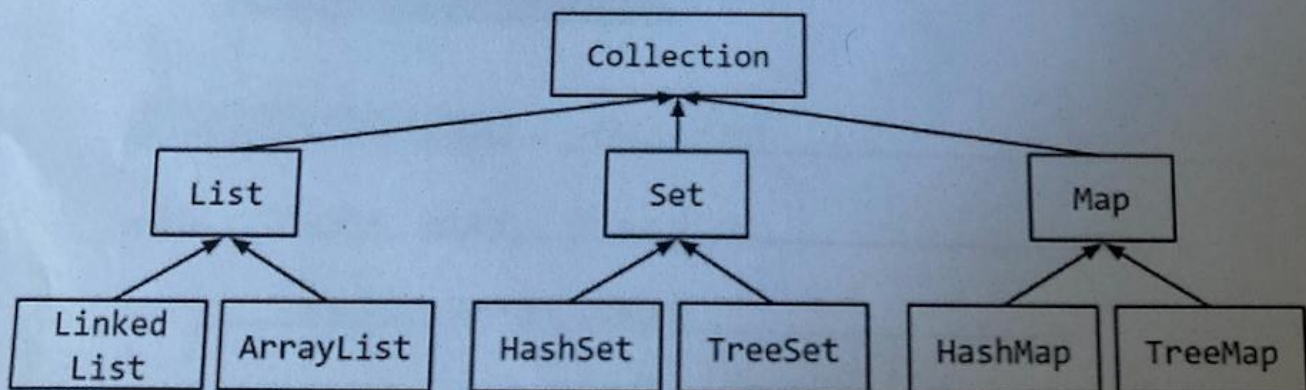
## String API

```
1 public class String {  
2  
3     /** Returns the character at the specified index (position) */  
4     public char charAt(int index) { ... }  
5  
6     /**  
7      * Returns a string that is a substring of this string. The substring begins at the specified  
8      * beginIndex and extends to the character at index endIndex - 1. Thus the length of the  
9      * substring is endIndex-beginIndex.  
10     */  
11     public String substring(int beginIndex, int endIndex) { ... }  
12  
13     /** Returns the length of this string. */  
14     public int length() { ... }  
15 }
```

## Map, Set, and List

```
1 public interface Map<K, V> ... { ...
2     boolean containsKey(K key);
3     V get(K key);
4     V getOrDefault(K key, V value);
5     void put(K key, V value);
6     Set<K> keySet();
7     Iterator<K> iterator();
8 }
9
10 public interface Set<K> ... { ...
11     boolean contains(K key);
12     void add(K key);
13     Iterator<K> iterator();
14 }
15
16 public interface List<T> ... { ...
17     boolean contains(T item);
18     void add(T item);
19     void add(int index, T item);
20     T get(int i);
21     T set(int i, T item);
22     int indexOf(Object o);
23     boolean remove(Object o);
24     Iterator<T> iterator();
25 }
```

Implementations:





# Problem 1

(6 Points) See Exam PDF for Problem Statement and instructions

```
public void rotateLeft (int k) {
```

```
    if ( k == 0 ) {  
        return ;  
    }
```

```
    int node tracker = this.first;
```

```
    while ( tracker.next != null ) {
```

```
        tracker = tracker.next;
```

```
    }
```

```
    tracker.next = this.first;
```

```
    _____ = _____;
```

```
    _____ = _____;
```

```
    rotateLeft ( k-1 );
```

```
}
```

## Problem 2

(15 Points) See Exam PDF for Problem Statement and instructions.

a)

ai)

$N$  is odd.  $n$  is even

$$\begin{array}{r} n-2 \\ \hline n-4 \\ \hline n-6 \end{array} \quad \downarrow \quad \frac{n}{2}$$

- ☐  $\Theta(1)$ 
☐  $\Theta(\log(\log N))$ 
☐  $\Theta(\log N)$ 
☐  $\Theta((\log N)^2)$ 
☒  $\Theta(N)$ 
☐  $\Theta(N \log N)$ 
☐  $\Theta(N^2 \log N)$ 
☐  $\Theta(N^3)$ 
☐  $\Theta(N^3 \log N)$ 
☐  $\Theta(N^4)$ 
☐  $\Theta(N^4 \log N)$ 
☐  $\Theta(N^5)$ 
☐ Worse than  $\Theta(3^N)$ 
☐ Never terminates (infinite loop)

aii)

$n^2$

$$\rightarrow 1 + 2 + 3 + \dots + n^2$$

$$= \frac{1+n^2}{2} \times n^2$$

- ☐  $\Theta(1)$ 
☐  $\Theta(\log(\log N))$ 
☐  $\Theta(\log N)$ 
☐  $\Theta((\log N)^2)$ 
☐  $\Theta(N)$ 
☐  $\Theta(N \log N)$ 
☐  $\Theta(N^2 \log N)$ 
☐  $\Theta(N^3)$ 
☐  $\Theta(N^3 \log N)$ 
☒  $\Theta(N^4)$ 
☐  $\Theta(N^4 \log N)$ 
☐  $\Theta(N^5)$ 
☐ Worse than  $\Theta(3^N)$ 
☐ Never terminates (infinite loop)

aiii)

- ☒  $\Theta(1)$ 
☐  $\Theta(\log(\log N))$ 
☐  $\Theta(\log N)$ 
☐  $\Theta((\log N)^2)$ 
☐  $\Theta(N)$ 
☐  $\Theta(N \log N)$ 
☐  $\Theta(N^2 \log N)$ 
☐  $\Theta(N^3)$ 
☐  $\Theta(N^3 \log N)$ 
☐  $\Theta(N^4)$ 
☐  $\Theta(N^4 \log N)$ 
☐  $\Theta(N^5)$ 
☐ Worse than  $\Theta(3^N)$ 
☐ Never terminates (infinite loop)



$\log_2 N$

- av)

$5 < N$ .  $\log n$ .

$$\log_2 N + i + \log_2 N + \dots + \log_2 N =$$

- $\Theta(1)$     $\bigcirc$   $\Theta(\log(\log N))$     $\bigcirc$   $\Theta(\log N)$     $\bullet$   $\Theta((\log N)^2)$     $\bigcirc$   $\Theta(N)$     $\bigcirc$   $\Theta(N \log N)$     $\bigcirc$   $\Theta(N^2)$   
 $\Theta(N^2 \log N)$     $\bigcirc$   $\Theta(N^3)$     $\bigcirc$   $\Theta(N^3 \log N)$     $\bigcirc$   $\Theta(N^4)$     $\bigcirc$   $\Theta(N^4 \log N)$     $\bigcirc$   $\Theta(2^N)$     $\bigcirc$   $\Theta(3^N)$   
 Worse than  $\Theta(3^N)$     $\bigcirc$  Never terminates (infinite loop)

b) See Exam PDF for Problem Statement and instructions.

bi)

See Exam PDF for Problem Statement and instructions.

```
public [redacted] [redacted] () {
    return this.coeffs.length - 1;
}
```

bii)

See Exam PDF for Problem Statement and instructions.

```
[redacted]
public [redacted] [redacted] ([redacted]) {
    return other.degree() - this.degree();
}
```

biii) (3 Points) See Exam PDF for Problem Statement and instructions.

```
public static [redacted] [redacted] ([redacted]) {
    [redacted] = new P.degree() + 1;
    new Coeffs[0] = 1;
    return new Polynomial (new Coeffs);
}
```



c) See Exam PDF for Problem Statement and instructions.

```
public class [redacted] {
```

```
    Polynomial pTest = Polynomial.simplify(p);
```

```
    int[] arr2 = new int[] {4, 3, 2, 1};
```

```
    Polynomial pTest2 = new Polynomial(arr2);
```

```
    assertTrue(pTest2.equals(p));
```

```
    Polynomial pTest3 = new Polynomial(new int[] {1, 0, 0, 0});
```

```
    assertTrue(pTest3.equals(pTest));
```

```
}
```

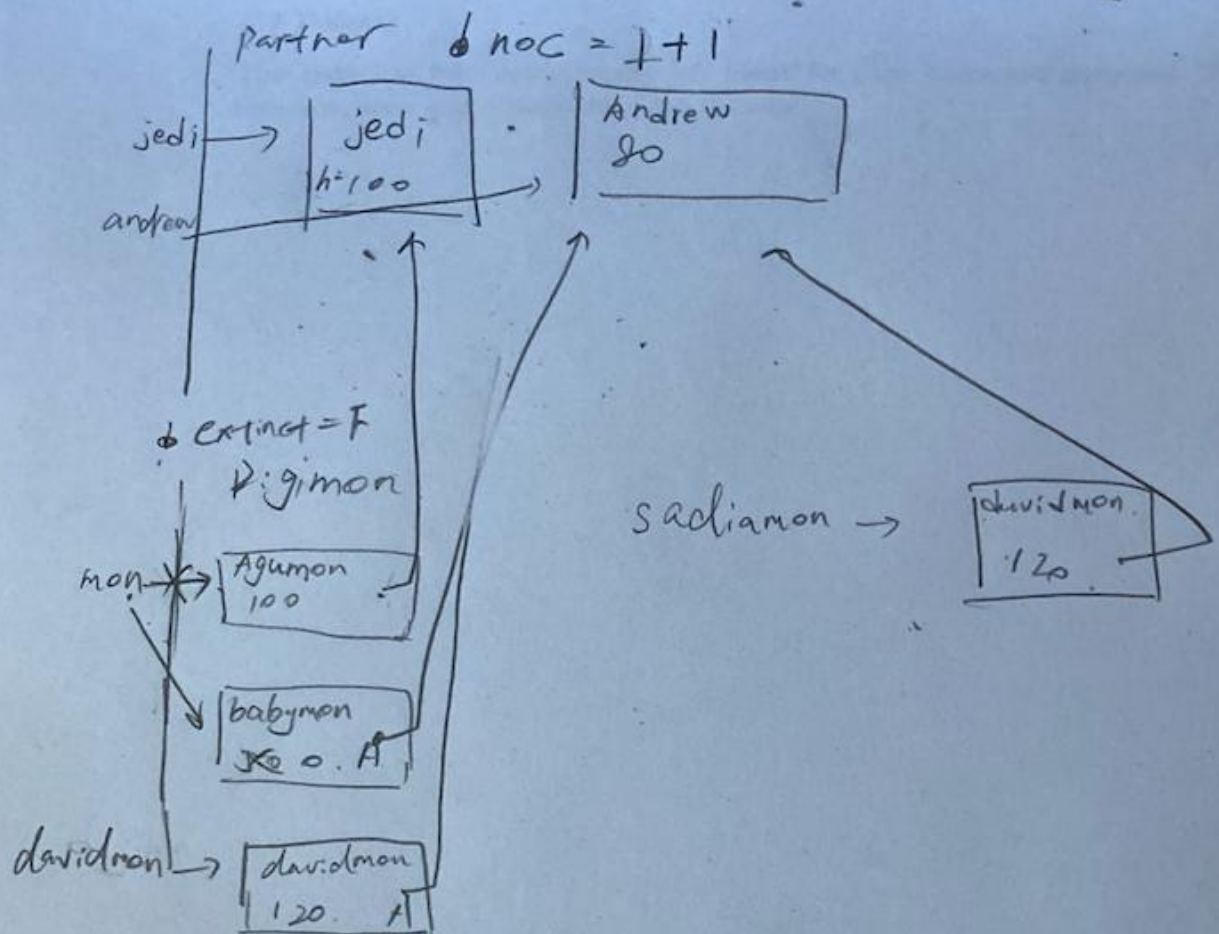
### Problem 3

(5 Points)

This page has been intentionally left blank for page alignment purposes. When you scan your answers template, scan and submit this page in order.



See Exam PDF for Problem Statement and instructions.



Line 1: Error

Line 2: 0

Line 3: Andrew

Line 4: 80

Line 5: Andrew

Line 6: 80

Line 7: true

Line 8: true

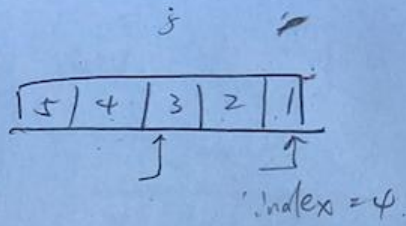
## Problem 4

(12 Points)

This page has been intentionally left blank for page alignment purposes. When you scan your answers template, scan and submit this page in order.



a) See Exam PDF for Problem Statement and instructions.



Line 1: false

Line 2: 2

Line 3: 3

Line 4: 21

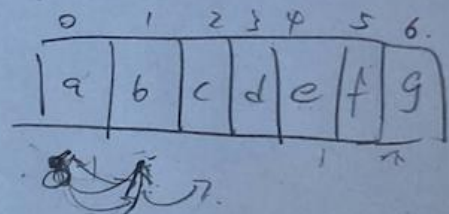
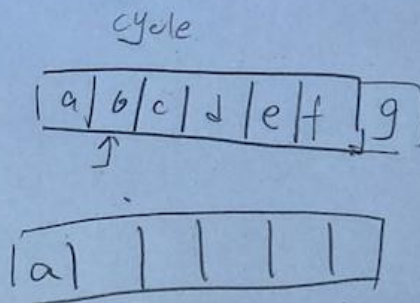
b) See Exam PDF for Problem Statement and instructions.

<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	

c) See Exam PDF for Problem Statement and instructions.

n is even, &lt; valid.

index = 1



i = 1

```

static [ ] [ ] ( [ ] ) {
    [ ] = new T[n];
    for ( [ ] ) {
        [ ] = cycle.next().prev();
        for ( [ ] ) {
            if ( i % 2 == 0 ) {
                last = cycle.next();
            } else {
                last = cycle.prev();
            }
        }
        out[i-1] = last;
    }
    return [ ];
}

```



## Problem 5

(16 Points)

This page has been intentionally left blank for page alignment purposes. When you scan your answer template, scan and submit this page in order.

a) See Exam PDF for Problem Statement and instructions.

```
    ( ) {  
        for (int x : arr) {  
            if (x != target) {  
                return false;  
            }  
        }  
        return true;  
    }
```

b) See Exam PDF for Problem Statement and instructions.

```
    ( ) {  
        for (int r = row-1; r ≤ row+1; r++) {  
            for (int c = col-1; c ≤ col+1; c++) {  
                if (r == row && c == col) {  
                    ;  
                }  
                if (r < 0 || r ≥ grid.length) {  
                    ;  
                }  
                if (c < 0 || c ≥ grid[0].length) {  
                    ;  
                }  
                if (grid[r][c]) {  
                    return ;  
                }  
            }  
        }  
        return ;  
    }
```



c) See Exam PDF for Problem Statement and instructions.

```

    ( ) {
        = new int[areas.length];
        = new int[area.length];
        = new int[area.length];
        for (int r=0; r < area.length; r++) {
            for (int c=0; c < area[r].length; c++) {
                if (stars[r][c]) {
                    int area = areas[r][c];
                    if (!starAdjacent(stars, r, c)) {
                        rowCounts[area-1] = r;
                        colCounts[area-1] = c;
                        areaCounts[area-1] = area;
                    }
                }
            }
        }
        return !areaCounts.contains(0) &&
               !colCounts.contains(0) &&
               !rowCounts.contains(0);
    }
}
```